**Practice Question -Fix Bugs**

**Project objective:**
**As a developer, fix the bugs in the application using the appropriate algorithms technique.**
**Background of the problem statement:**
**You have been assigned a few tasks during the sprint planning. Solving the bugs raised**
**by the testing team is one among them. You are given the boilerplate code and are asked**
**to complete it by fixing the bugs.**
**STEP 1: Create a Java project in your IDE.**
**STEP 2: Select new Java class-provide class name follow camelCasing).**
**STEP:3 Inside the search expenses,I have created a variable to get the search element**
**from user.**
**STEP:4 Then in if condition if condition if the arraylist contains the search element then**
**it will print the element you searched for is found.**
**STEP:5 If the condition is not true tyen it will print the element is not found.**
**STEP:6 Then for sorting the values,I have used comparator.sort() method to sort the**
**element.**
**STEP:7 Then by using while condition, I have checked the values using true that is if the**
**values entered if it is correct then it will execute next .**
**STEP:8 Then I have used System.exit(1) method to stop in case 6 or else the loop will not**
**stop because of while condition.Execute the program.**

**Code:**

```
package arthematicoperations;

import java.util.ArrayList;
```

```
import java.util.Scanner;

public class FixBugs {

    public static void main(String[] args) {
        /*System.out.println("Hello World!");*/
        System.out.println("\n***************************************\n");
        System.out.println("\tWelcome to TheDesk \n");
        System.out.println("***************************************");
        optionsSelection();

    }
    private static void optionsSelection() {
        String[] arr = {"1. I wish to review my expenditure",
                "2. I wish to add my expenditure",
                "3. I wish to delete my expenditure",
                "4. I wish to sort the expenditures",
                "5. I wish to search for a particular expenditure",
                "6. Close the application"
        };
        int[] arr1 = {1,2,3,4,5,6};
        int  slen = arr1.length;
        for(int i=0; i<slen;i++){
            System.out.println(arr[i]);
            // display the all the Strings mentioned in the String array
        }
        ArrayList<Integer> arrlist = new ArrayList<Integer>();
        ArrayList<Integer> expenses = new ArrayList<Integer>();
        expenses.add(1000);
        expenses.add(2300);
        expenses.add(45000);
        expenses.add(32000);
        expenses.add(110);
        expenses.addAll(arrlist);
        System.out.println("\nEnter your choice:\t");
        Scanner sc = new Scanner(System.in);
        int  options =  sc.nextInt();
```

```
        for(int j=1;j<=slen;j++){
            if(options==j){
                switch (options){
                    case 1:
                        System.out.println("Your saved expenses are listed below: \n");
                        System.out.println(expenses+"\n");
                        optionsSelection();
                        break;
                    case 2:
                        System.out.println("Enter the value to add your Expense: \n");
                        int value = sc.nextInt();
                        expenses.add(value);
                        System.out.println("Your value is updated\n");
                        expenses.addAll(arrlist);
                        System.out.println(expenses+"\n");
                        optionsSelection();

                        break;
                    case 3:
                        System.out.println("You are about the delete all your expenses!
\nConfirm again by selecting the same option...\n");
                        int con_choice = sc.nextInt();
                        if(con_choice==options){
                              expenses.clear();
                           System.out.println(expenses+"\n");
                           System.out.println("All your expenses are erased!\n");
                        } else {
                           System.out.println("Oops... try again!");
                        }
                        optionsSelection();
                        break;
                    case 4:
                        sortExpenses(expenses);
                        optionsSelection();
                        break;
                    case 5:
                        searchExpenses(expenses);
```

```
                optionsSelection();
                break;
            case 6:
                closeApp();
                break;
            default:
                System.out.println("You have made an invalid choice!");
                break;
        }
    }
  }

}
private static void closeApp() {
   System.out.println("Closing your application... \nThank you!");
      }
private static void searchExpenses(ArrayList<Integer> arrayList) {
     Scanner in=new Scanner(System.in);
   int leng = arrayList.size();
   System.out.println("Enter the expense you need to search:\t");
   //Complete the method
   int key=in.nextInt();
   int found=0;
   int index=0;
   for(int i=0; i<leng; i++)
      if(arrayList.get(i)==key) {
         found=1;
         index=i;
      }
   if(found==1) {
    System.out.println(key+ " is found at index " +index);
   }
}
private static void sortExpenses(ArrayList<Integer> arrayList) {
   int arrlength =  arrayList.size();
  //Complete the method. The expenses should be sorted in ascending order.
   for (int i = 0; i < arrlength ; i++) {
```

```
            for (int j = arrlength - 1; j > i; j--) {
                if (arrayList.get(i) > arrayList.get(j)) {
                    int temp = arrayList.get(i);
                    arrayList.set(i,arrayList.get(j)) ;
                    arrayList.set(j,temp);
                }
            }
        }
        for (int i: arrayList) {
            System.out.print(i+ " ");
        }
        System.out.println();
    }

}
```